Technical Report

Automating Microsoft SQL Server 2012 Deployment and Storage Provisioning

Microsoft Solutions Integration Group, NetApp

September 2012 | TR-4105

TABLE OF CONTENTS

LIST OF FIGURES

1 Automating Microsoft SQL Server 2012 Deployment and Storage Provisioning Overview

Automating Microsoft® SQL Server® 2012 and storage provisioning on NetApp® storage is facilitated through the use of virtualization of the SQL Server instances and through the use of NetApp storage management features (storage efficiency, data protection, cloning, Windows PowerShell™ support). The features deliver the following benefits:

- Rapid repeatable process to deploy SQL Server database instances and databases on NetApp storage
- Improved quality of deployed SQL Server instances and associated storage
- Reduce administrative overhead to support SQL Server deployment process
- Advanced storage management, provisioning, backup, and data recovery features

For additional information and recommendations for virtualizing SQL Server 2012, refer to Microsoft TechNet, http://technet.microsoft.com/en-us/library/ms143506, and Microsoft Support, http://support.microsoft.com/?id=956893.

1.1 Purpose and Scope

Automation and virtualization of SQL Server on NetApp storage provide significant reductions in cost and increased business agility. This report is designed to help reduce customer and partner deployment times by providing specific guidance for provisioning virtualized SQL Server instances on NetApp storage.

1.2 Target Audience

This paper is recommended for the following audiences:

- SQL Server database administrators
- Private cloud architects
- NetApp field personnel and partners

1.3 Technology Solution

This packaged solution is a tightly integrated software management stack based on Microsoft Windows Server® components, SQL Server 2012, and NetApp storage management components. The goal is to provide an integrated management experience that allows customers to easily deploy virtualized SQL Server instances in both production and test environments.

1.4 Use Case Summary

Automation is a key theme among SQL Server administrators, and the addition of the Windows PowerShell components in SQL Server 2012 enables administrators to automate almost every aspect of their daily operations.

This package is designed to automate the deployment of SQL Server 2012 in an enterprise setting and also for lab environments. The key features that this platform offers are:

- **Easier deployment of SQL Server 2012.** Easier deployment of SQL Server is enabled by the use of Windows PowerShell scripts that automate the cloning of SQL Server virtual machines from a gold master copy containing the desired Windows Server operating system version and SQL Server 2012 binaries. Prior to the script being run, a gold master virtual machine stored on a VMware® VMDK file or a Hyper-V™ VHD file will have to be created and ready for use.
- **Storage provisioning.** Storage provisioning is provided by the NetApp Data ONTAP® PowerShell Toolkit 2.1 cmdlets. This allows for consistency of operations and permits the solutions team to

provide operational procedures based on a consistent architecture. All operational procedures are developed through the use of these tools. Wherever possible, NetApp storage efficiency and data protection technologies are used as key capabilities of the storage offerings.

- **NetApp SnapDrive for Windows installation and configuration.** SnapDrive® for Windows® (SDW) installation is accomplished using Windows PowerShell and Windows batch files. The SnapDrive application helps with storage provisioning and managing disks in both physical and virtual environments. SDW manages the LUNs on the storage system, making them available as local disks on Windows hosts.
- **NetApp SnapManager for SQL Server installation.** NetApp SnapManager® for SQL Server installation is also triggered using the Windows PowerShell and Windows batch files. SnapManager for SQL Server (SMSQL) provides an integrated data management solution for Microsoft SQL Server 2012 that enhances the availability, scalability, and reliability of SQL Server databases. SMSQL provides rapid online backup and restoration of databases, along with local or remote backup set mirroring for disaster recovery.
  SMSQL uses online Snapshot™ technologies that are part of Data ONTAP to integrate with SQL Server backup and restore APIs and the Volume Shadow Copy Service (VSS). SMSQL also uses SnapMirror® to support disaster recovery for SQL Server 2012 environments that are not using availability group designs for disaster recovery.

  SMSQL provides the following data management capabilities:

  - Migrating SQL Server databases and transaction logs to NetApp LUNs, VMware RDM LUNs, VMware VMDK over NFS virtual disks, and Hyper-V pass-through LUNs
  - Backing up SQL Server databases and transaction logs stored on NetApp LUNs/VMDKs
  - Verifying SQL Server databases and transaction logs in backup sets
  - Cloning SQL Server databases from SMSQL backup sets
  - Managing SMSQL backup sets
  - Archiving SMSQL backup sets
  - Restoring SQL Server databases and transaction logs from previously created backup sets

2 Architecture Primary Use Case

NetApp solutions are designed to be tightly integrated with Microsoft SQL Server 2012. Running Microsoft SQL Server 2012 in a virtualized environment with NetApp enables better availability, flexibility, and more efficient use of infrastructure with no impact on performance. By using an automated and virtualized environment, IT organizations can make much more effective use of their server and storage infrastructure.

The combination of virtualization and NetApp solution provides an architectural design that facilitates consolidation:

- Virtualization platforms can simultaneously support VMs running different operating systems on the same physical server.
- All NetApp systems, from entry level to high end, are based on a single unified storage architecture that simplifies management with common tools and processes for both physical and virtual environments. A single NetApp system supports major storage protocols, including Fibre Channel and iSCSI, and both Fibre Channel and SATA disk drives, and can be used to consolidate all data types.

By using NetApp as the storage infrastructure, the virtualized environment benefits from the advanced storage management, provisioning, backup, and data recovery features of NetApp as well as significantly greater storage efficiency. NetApp allows you to add storage as needed by running under a hypervisor platform for added efficiency, and you can quickly provision additional SQL Server instances.

SQL Server 2012 virtualization allows for the efficient use of hardware resources, which can be combined along with the other key areas of server virtualization, including better availability, lower cost, and increased flexibility. Multiple benefits can be realized by using SQL Server 2012 in a virtualized environment with NetApp storage technology, including:

- **Effective usage of server hardware.** Migrating the entire environment from dedicated physical servers that have fairly low utilization rates can lead to significantly higher server utilization.
- **Infrastructure savings.** Save on power and space.
- **Automate deployment of SQL Server 2012 virtual machines.** Deployment of SQL Server 2012 virtual machines becomes easier and more rapid and has higher quality through the use of virtual machine templates backed by NetApp gold master images.
- **Automate provisioning of SQL Server 2012 storage and databases.** Provisioning databases and storage for SQL Server 2012 becomes easier and more rapid and has higher quality through the use of the NetApp PowerShell Toolkit, Windows PowerShell, and SQL Server 2012 Transact SQL Server and Windows PowerShell commands to automate the process.
- **Storage capacity savings.** NetApp storage deduplication of SQL Server virtual machines can result in large storage capacity savings and increased storage efficiency.

2.1 Server

The host server architecture is a critical component of the virtualized infrastructure as well as a key variable in the consolidation ratio and cost analysis.

The system architecture of the host server refers to the general category of the server hardware itself. The primary factor to consider when selecting system architectures is that each virtualization host can contain multiple guests with multiple workloads. The critical factors are processor, RAM, storage, and network capacity; high I/O capacity; and low latency. The host server must be able to provide the required capacity in each of these categories.

2.2 Storage

The storage design for any virtualization-based solution is a critical element that is typically responsible for a large percentage of the solution’s overall cost, performance, and agility.

Although many storage options exist, organizations should choose their storage devices based on their specific data management needs. Storage devices typically include modular, flexible midrange SANs and high-end SANs. Modular midrange SANs are procured independently and can be chained together to provide greater capacity. NetApp FAS series controllers are efficient, can grow with the environment as needed, and require less up-front investment than high-end SANs.

Large enterprises might have larger storage demands and might need to serve a larger set of customers and workloads. In this case, high-end SANs can provide the highest performance and capacity. High-end SANs typically include more advanced features such as continuous data availability through technologies such as replication and clustering. The NetApp FAS series unified architecture has all of these capabilities—including replication and clustering support for all FAS controllers—from the smallest 2000 series to the largest 6300 series controller. The FAS controllers used in this solution should be licensed for the full range of NetApp Data ONTAP features.

2.3 Virtualization

Virtualization is based on the abstraction of physical system resources so that multiple logical partitions can be created and can host a wide range of operating systems that run simultaneously on a single physical server. Rather than paying for many underutilized servers and direct-attached storage solutions, each dedicated to a specific workload, server virtualization in conjunction with SANs allows those workloads to be consolidated onto a smaller number of more efficiently utilized physical systems. Server virtualization provides the following benefits:

- Consolidates multiple underutilized physical servers on a single host running VMs
- Reduces workforce, space, and kilowatts by taking advantage of virtualization for server consolidation and agility

Virtualization can also help to simplify and accelerate provisioning of virtual machines. The vast majority of virtualized SQL Server 2012 architectures are deployed on Hyper-V and VMware virtualization platforms. Please refer to the Microsoft TechNet article at http://technet.microsoft.com/en-us/library/ms143506 during the planning phase before using the scripts in this paper for deployment into production or test environments.

3 NetApp Storage Efficiency

In simple terms, storage efficiency means increasing storage utilization and decreasing storage costs. This guide implements the following technologies to enhance storage efficiency and to optimize the existing storage in the infrastructure as well as deferring or avoiding future storage needs:

- **RAID-DP.** RAID-DP® is the NetApp implementation of double-parity RAID 6, which is an extension of the original NetApp Data ONTAP WAFL® RAID 4 design. Unlike other RAID technologies, RAID-DP provides the ability to achieve a higher level of data protection without any performance effect while consuming a minimal amount of storage.
- **SATA.** The performance acceleration provided by WAFL and the double-disk protection provided by RAID-DP make economical, large-capacity SATA drives practical for production application use. In addition, to negate the read latencies associated with large drives, SATA drives can be used with the NetApp Flash Cache card, which significantly increases performance with large working set sizes.
- **Snapshot capability.** NetApp Snapshot technology provides zero-cost, near-instantaneous backup, point-in-time copies of the volume or LUN by preserving Data ONTAP WAFL consistency points (CPs). Creating Snapshot copies incurs minimal performance effect because data is not moved.
- **NetApp deduplication.** The deduplication process stores only unique blocks of data in the volume and creates additional metadata in this process.

The core enabling technology of deduplication is fingerprints. When deduplication runs for the first time on a FlexVol® volume, it scans the blocks and creates a fingerprint database that contains a sorted list of all fingerprints for used blocks in the flexible volume. Each 4KB block in the storage system has a digital fingerprint, which is compared to other fingerprints on the volume. If two fingerprints are found to be the same, a byte-for-byte comparison is done of all bytes in the block. If they are an exact match, the duplicate block is discarded, and the space is reclaimed.

Deduplication consumes system resources and can alter the data layout on disk. Due to the application I/O pattern and the effect of deduplication on the data layout, read and write I/O performance can vary.

Note: Deduplication is transparent to SQL Server 2012, and the block changes are not recognized by SQL Server. The SQL Server 2012 database remains unchanged in size from the host’s perspective, even though capacity savings occur at the NetApp flexible volume level.

For best practices on how to configure Microsoft SQL Server 2012 for 7-Mode storage systems, refer to the Microsoft SQL Server and SnapManager for SQL Server on NetApp Storage Best Practices Guide, http://media.netapp.com/documents/tr-4003.pdf.

4 Data Protection

Backup

This solution provides a means of data backup. NetApp SnapManager for SQL Server allows the SQL Server data to be backed up by using NetApp native Snapshot technology. The integration of NetApp

Snapshot technology and Volume Shadow Copy services APIs is accomplished using the Data ONTAP VSS hardware provider that is installed on SQL Server. The SnapManager for SQL Server software leverages the SQL Server VDI to freeze the SQL Server databases being backed up so that a consistent Snapshot copy of the SQL Server databases can be created.

5 Storage Layout Planning

5.1 Aggregate Recommendations

The deployment process assumes that the SQL Server virtual machines will be deployed onto a separate aggregate from the SQL Server data aggregate. The deployment process also assumes that the SQL Server data aggregate adheres with NetApp best practices for Microsoft SQL Server 2012 with respect to the aggregate planning.

Pooling all of the available disks into a single, large aggregate might maximize performance; however, it might not meet the data availability requirements for the SQL Server services. Creating separate aggregates for SQL Server database volumes can help isolate the loss of SQL Server services to databases hosted on the affected aggregate. If SQL Server 2012 is deployed in an availability group configuration, placing the database copies on separate aggregates will keep SQL Server services available in the unlikely event that an aggregate is lost.

For best practices on how to configure Microsoft SQL Server 2012 for 7-Mode storage systems, refer to the Microsoft SQL Server and SnapManager for SQL Server on NetApp Storage Best Practices Guide, http://media.netapp.com/documents/tr-4003.pdf.

5.2 Volume Planning

In this solution, separate NetApp flexible volumes are created for each SQL Server database and transaction log LUN. Each SQL Server instance will have its own database volumes. SQL Server instances will not share NetApp flexible volumes or NetApp LUNs.

Data ONTAP enables the creation of flexible (FlexVol) volumes for managing data without the need to assign physical disks to the volumes. Instead, the FlexVol volumes enjoy performance benefits from a larger pool of physical disks called an aggregate. This results in the following additional benefits for Microsoft SQL Server environments:

- A large number of volumes can be created, all with independent Snapshot copy schedules, mirroring policies, and so on.
- All volumes can be managed independently while receiving the maximum I/O benefit of a much larger pool of disks.

Volume layout is critical in creating and sustaining a highly available SQL Server environment. Careful consideration of various backup groups, disaster recovery scenarios, and even archiving solutions helps determine the placement of volumes onto aggregates and the placement of the corresponding LUNs onto those volumes.

For best practices on how to configure Microsoft SQL Server 2012 for 7-Mode storage systems, refer to the Microsoft SQL Server and SnapManager for SQL Server on NetApp Storage Best Practices Guide, http://media.netapp.com/documents/tr-4003.pdf.

5.3 LUN Layout Recommendations

When considering SQL Server 2012 LUN configuration, the number and size of LUNs you should provision largely depend on your recovery time objectives (RTOs) for the databases that will be stored on the LUNs.

Databases that share a LUN can have slower RTO times because a single database restore in a multiple databases per LUN scenario is a streaming restore where the database files are copied back into the active file system from a Snapshot copy.

Databases that do not share a LUN have faster RTO times because the restore operation for the database is at the LUN level where the LUN is restored from a Snapshot copy.

Two LUNs per Database

In this solution, each SQL Server database and transaction log set is placed on a separate LUN. This solution provides greater flexibility in terms of recovery options but increases the total number of LUNs required.

One LUN per Volume

In this solution each LUN is placed in its own volume. This solution provides greater flexibility in terms of backup and recovery options, but increases the total number of volumes required.

For best practices on how to configure Microsoft SQL Server 2012 for 7-Mode storage systems, refer to the Microsoft SQL Server and SnapManager for SQL Server on NetApp Storage Best Practices Guide, http://media.netapp.com/documents/tr-4003.pdf.

6 Capacity Planning

When you design a SQL Server 2012 virtualized environment, the SQL Server design principles remain basically the same as they would be for nonvirtualized environments, if the SQL Server data is stored on NetApp LUNs.

Note: For SQL Server database data stored on VMDKs over NFS or on CIFS/SMB shares, the storage design is quite different and is out of the scope of this document.

The environment must be designed to meet the desired performance, reliability, and capacity requirements. Design considerations such as how the databases will be used (for example, OLTP, data warehouse, and so on) play a major role in the storage design for SQL Server.

Because virtualization provides the flexibility to make changes to the design of the environment at a later stage, certain organizations might spend less time on their design during the planning stage. As a best practice, spend enough time designing the environment to avoid any pitfalls after the implementation.

A properly sized SQL Server 2012 environment meets both the performance and storage capacity requirements to satisfy the business requirements for the databases (SLAs). You should use the following primary tools when planning a SQL Server environment for a customer:

- Microsoft Windows Performance Monitor
- NetApp Storage Performance Modeler (SPM) Tool
- NetApp consulting systems engineer

The sizing information provided by these resources is an important component for planning a SQL Server 2012 environment and provides a framework for database layout and LUN requirements. When sizing SQL Server 2012 deployments using NetApp storage, it is important to use the NetApp System Performance Modeler and validate the storage design with a NetApp consulting systems engineer specializing in Microsoft SQL Server.

Note: Use the previously mentioned tools and size the SQL Server environment before using the scripts to deploy the SQL Server instances and provision storage.

7 Storage Automation

The main objectives of automating the deployment of virtualized SQL Server instances and SQL Server storage are:

- Reduce the total cost of ownership (TCO) of the SQL Server environment
- Increase business agility through a rapid, repeatable, and reliable deployment process
- Improve the quality of SQL Server deployment and storage configuration

Windows Server 2008 R2 allows for the automation of system management using Windows PowerShell. SQL Server 2012 allows for the automation of SQL Server management functions using Windows PowerShell called SQL Server PowerShell (http://msdn.microsoft.com/en-us/library/hh245198) and also using the legacy Transact-SQL commands. VMware allows for the automation of VMware management functions using Windows PowerShell called PowerCLI. NetApp also supports storage management using Windows PowerShell called the NetApp PowerShell Toolkit.

All four Windows PowerShell frameworks allow for the orchestration of virtualized SQL Server deployment tasks. Microsoft System Center Orchestrator and Cloudera are examples of orchestration platforms that can call Windows PowerShell scripts embedded in orchestrator workflows to help achieve the goals of automating SQL Server deployment tasks.

7.1 Considerations

Infrastructure Deployment

When deploying SQL Server, choosing the deployment model is a critical decision. Choices range from manual installation, which is highly inefficient, to varying degrees of automation, up to enterprise-class management systems. To achieve the architecture principle of predictability, all infrastructure components should be able to be deployed and configured in a repeatable and automated fashion.

The key components for successful deployment are the standard Windows Server roles such as Active Directory® Domain Services and Domain Name System (DNS). Using these technologies, it is possible to provide a robust deployment infrastructure using standard in-box solutions and toolkits.

The scope of the process and automation detailed in this document is based on the following assumptions:

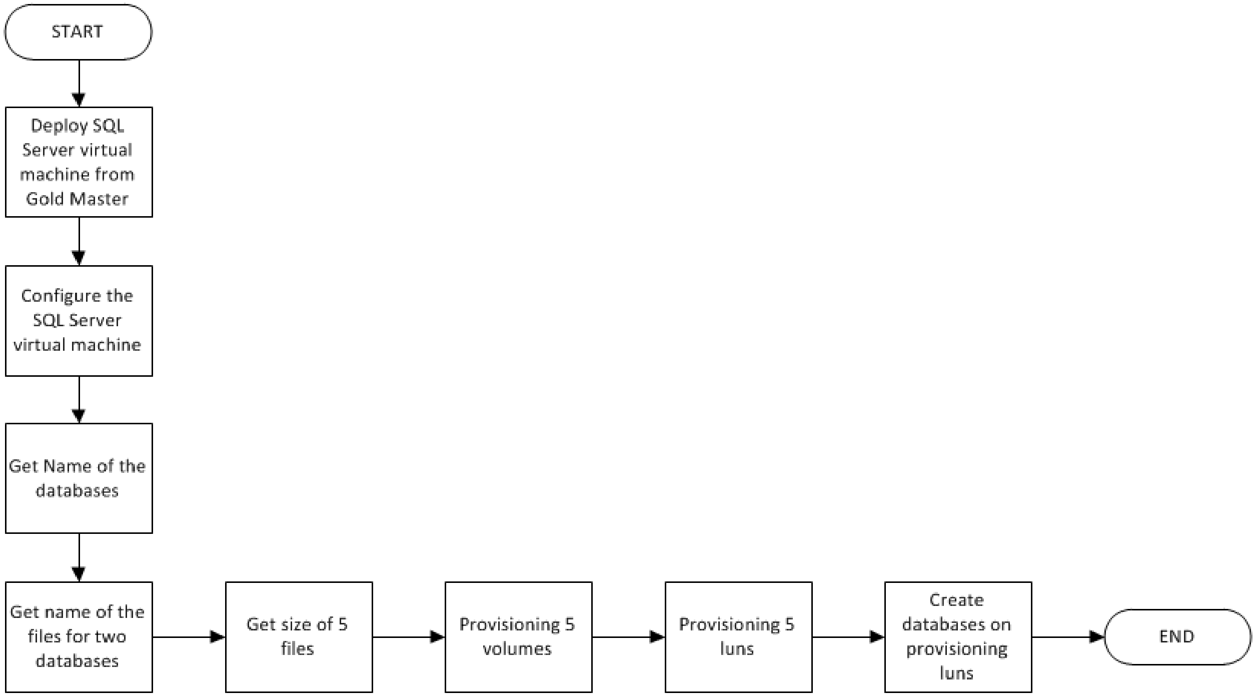
- Only VMware virtualization is used.
- A VMware gold master SQL Server virtual machine (VMDK) image has been created.
- The VMware NFS datastore on NetApp storage has been provisioned and correctly configured.
- The gold master virtual machine image is configured per the SQL Server 2012 hardware and software system requirements documented in Microsoft TechNet at http://technet.microsoft.com/en-us/library/ms143506.
- The gold master SQL Server virtual machine image is installed with Windows Server 2008 R2 Service Pack 1 Enterprise Edition.
- The gold master SQL Server virtual machine image is installed with SQL Server 2012 Enterprise Edition.
- The NetApp Data ONTAP PowerShell toolkit is installed on the gold master SQL Server virtual machine image. You can download the toolkit here. Refer to TR-3896: NetApp PowerShell Survival Guide, http://media.netapp.com/documents/tr-3896.pdf, for more information on how to use Windows PowerShell and the toolkit to manage NetApp storage.
- The SQL Server instance is a standalone instance of SQL Server.

- The provisioning process will use SnapDrive installed on the provisioned SQL Server virtual machine to provision iSCSI LUNs connected to the SQL Server virtual machine to store the SQL Server database data.
- The database provisioning process will offer the option of deploying multiple databases on a single LUN or a single database on a single LUN.

8 Automating SQL Server Deployment and Database Provisioning Process

Figure 1 illustrates the workflow for the SQL Server deployment and database provisioning process.

Figure 1) SQL Server deployment process.


START
Deploy SQL Server virtual machine from Gold Master
Configure the SQL Server virtual machine
Get Name of the databases
Get name of the files for two databases
Get size of 5 files
Provisioning 5 volumes
Provisioning 5 luns
Create databases on provisioning luns
END

8.1 The Workflow

Before you begin, make sure the virtualization environment is properly connected to the NetApp storage and the gold master virtual machine is built and stored on the NetApp storage.

The steps shown in **Error! Reference source not found.** are the sequence of steps involved in Microsoft SQL Server 2012 deployment and storage provisioning:

1. Create a clone of the SQL Server virtual machine (VMDK) from the gold master image.
2. Start the virtual machine and run through out-of-the-box setup experience. Log in to the SQL Server virtual machine, configure the LAN and iSCSI network settings, and then add SQL Server virtual machine to the Active Directory domain.
3. Start the iSCSI service, install SnapDrive for Windows, and rename the SQL Server instance.
4. Create the volumes, LUNs, and databases for the SQL Server instance.
5. Install SnapManager for SQL Server.

8.2 Provisioning Scripts

Two scripts contain the various functions that are used for automation of virtualized SQL Server instances, SQL Server storage provisioning, and SMSQL installation.

The scripts are as follows:

- Pre-Provisioning.ps1
- ProvisioningSQL.ps1
- SMSQLInstall.ps1

Prerequisites

Take the following actions before running the scripts.

1. Download SnapDrive and SnapManager software installation packages.
2. Provide the necessary parameters that are required by the scripts.
3. Confirm that the execution policy for scripts is set to “Unrestricted.”
4. Use Windows PowerShell or Windows PowerShell ISE to execute the scripts.

The following subsections elaborate on the scripts mentioned in this section.

Pre-Provisioning.ps1

This script automates the process of configuring the iSCSI service, installs SnapDrive for Windows, and renames the SQL Server instance on the SQL Server virtual machine.

ProvisioningSQL.ps1

This script creates and configures the flexible volumes on the NetApp storage for the database and transaction log data, uses SnapDrive for Windows to create the database and transaction log LUNs, and then creates the database and transaction log files on the LUNs.

SMSQLInstall.ps1

This script installs and configures the SnapManager for SQL Server application.

8.3 Logging

The installation logs for the SnapDrive and SnapManager installation packages are located in the same directory as the SnapDrive and SnapManager installation packages.

9 Appendix

Supporting Documents

- NetApp Hyper-V Cloud Fast Track with Cisco:
  www.netapp.com/us/library/white-papers/wp-7132.html
- NetApp Solution for Private Clouds:
  http://media.netapp.com/documents/rp-microsoft-private-clouds.pdf
- Microsoft SQL Server and SnapManager for SQL Server Best Practices Guide:
  http://media.netapp.com/documents/tr-4003.pdf
- NetApp SnapDrive for Windows Documentation:
  https://support.netapp.com/documentation/productlibrary/index.html?productID=30049

- NetApp SnapManager for SQL Server Documentation
  http://support.netapp.com/documentation/productlibrary/index.html?productID=30041

Scripts

SQL Server Virtual Machine Preparation (Pre-Provisioning.ps1)

```
# +--------------------------------------------------------------------------
# | File : Pre-Provisioning.ps1
# | Version : 1.00
# | Purpose : Prepare server before provisioning databases
# | Synopsis:
# | Usage : .\Pre-Provisioning.ps1
# +---------------------------------------------------------------------------
# | Maintenance History
# | -------------------
# | Name           Date   Version  C/R  Description
# | ----------------------------------------------------------------------------------
# | Pat Sinthusan            2012-0-27   1.00    Script originated for VMWorld 2012
# +-------------------------------------------------------------------------------
# *******************************************************************************

#Start iSCSI Service
$ServiceName = “MSiSCSI”
$Service = Get-Service –name $ServiceName
if ($Service.Status –ne “Running”) {Start-Service $ServiceName}
$Service.Refresh()
while ($Service.Status -ne "Running"){
        $Service.Refresh()
        Start-Sleep -Seconds 5
}
Add-NaHostIscsiTargetPortal
Get-NaHostIscsiTarget (Get-NaIscsiNodeName) | Connect-NaHostIscsiTarget

$Account = Read-Host "Please provide SnapDrive Service Account"
$Password = Read-Host "Please provide SnapDrive Service Account Password"
$SDInstallDir = Read-Host “Please provide directory where Snapdrive Installation Package is located”
#Install SnapDrive
```

SQL Server Virtual Machine Preparation (Pre-Provisioning.ps1)

```
$cmd = $SDInstallDir + '\SnapDrive.exe /s /v /qb+ SILENT_MODE=1 /Li C:\' + $SDInstallDir +
'\SDInstall.log INSTALLDIR=\"c:\Program Files\NetApp\SnapDrive\" '

$cmd = $cmd + 'SVCUSERNAME=' + $Account + ' SVCUSERPASSWORD=' + $Password + '
SVCCONFIRMUSERPASSWORD=' + $Password

$cmd = $cmd + ' SDW_WEBSRV_TCP_PORT=808 SDW_WEBSRV_HTTP_PORT=4098
TRANSPORT_PRT_SELECTION=3 TRANSPORT_PRT_PORT=443 '

$cmd = $cmd + ' TRANSPORT_PROTOCOL_LOGON_USERNAME=root
TRANSPORT_PROTOCOL_LOGON_PASSWORD'

$cmd

Invoke-Command $cmd


#Drop old server name

$cmd = "declare @old sysname; set @old = (select name from sys.servers); exec sp_dropserver @old"

sqlcmd -Q $cmd

#Add new server name

$cmd = "declare @new sysname; set @new = (select convert(sysname, (select
SERVERPROPERTY('ComputerNamePhysicalNetBIOS')))); exec sp_addserver @new, 'local'"

sqlcmd -Q $cmd

#Restart server to make iSCSI take effect.

Shutdown /f /r
```

Storage and Database Provisioning (ProvisioningSQL.ps1)

```
# +--------------------------------------------------------------------------

# | File : ProvisioningSQL.ps1

# | Version : 1.00

# | Purpose : Provisioning 2 databases

# | Synopsis:

# | Usage : .\ProvisioningSQL.ps1

# +---------------------------------------------------------------------------

# | Maintenance History

# | -------------------

# | Name           Date   Version  C/R  Description

# | --------------------------------------------------------------------------------

# | Pat Sinthusan           2012-0-27   1.00    Script originated for VMWorld 2012

# +------------------------------------------------------------------------------

# ********************************************************************************
```

Storage and Database Provisioning (ProvisioningSQL.ps1)

```
powershell.exe -ExecutionPolicy Unrestricted -NoLogo -NonInteractive
#Need to do this first otherwise all varaibles will get wipe out.
if ((Get-Module| select -exp name) -notcontains 'SQLPS') {Import-Module SQLPS -DisableNameChecking}

$DB1 = @{}
$DB2 = @{}

function GetDatabaseName($position, $Database){
    $DatabaseName = Read-Host "Please provide name of database number $position ($Database)"
    if ($DatabaseName -eq "") {$DatabaseName = $Database}
    return $DatabaseName
}
function GetDatabaseFile($FileType, $Database, $Default){
    $filename = Read-Host "Please provide $FileType file name for $Database database ($Default)"
    if ($FileType -ne "ldf"){
        if ($filename -eq ""){$filename = $Database + "." + $FileType} else {$filename = $filename + "." + $FileType}
    } else  {
        if ($filename -eq ""){$filename = $Database + "_log." + $FileType} else {$filename = $filename + "_log." + $FileType}}
    return $filename
}

function GetFileSize($FileName, $Default){
    $FileSize = Read-Host "Please provide file size of $FileName in GB ($Default)"
    if ($FileSize -eq "") {$FileSize = $Default}
    return $FileSize
}

function GetVolumeName ($filename, $Default){
    $volname = Read-Host "Please provide volume name for $filename ($Default)"
    if ($volname -eq "") {$volname = $Default}
    return $volname
}
```

Storage and Database Provisioning (ProvisioningSQL.ps1)

```
function GetLunName ($filename, $Default){
        $lunname = Read-Host "Please provide lun name for $filename ($Default)"
        if ($lunname -eq "") {$lunname = $Default}
        return $lunname
}

function ConnectController{
        $ControllerName = Read-Host "Please provide controller name or ip"
        if ($ControllerName -eq ""){
                Write-Output "Controller name or ip need to be specified"
        }
        if ((Get-Module| select -exp name) -notcontains 'DataOntap') {Import-Module DataOntap}
        $Controller = Connect-naController $ControllerName -cred root
        if ($Controller -eq $null) {
                Write-Output "Cannot connect to controller!"
                Exit
        }
        $Controller
}

function GetAggregateName{
        $AggregateName = Read-Host "Please provide Aggregate name"
        if ($AggregateName -eq $null) {
                Write-Output "Aggregate name need to be provided!"
                Exit
        }
        return $AggregateName
}

function CreateVolume($AggregateName, $Vol, $Size){
        New-naVol $Vol $AggregateName ($Size * 1.5)
}
```

Storage and Database Provisioning (ProvisioningSQL.ps1)

```
function CreateLun ($MountPoint, $Controller , $volname, $lunname, $computername, $igroup, $Size){
        $Size = $Size * 1.2
        sdcli disk create -d $MountPoint -p $Controller`:/vol/$volname/$lunname -IG $Computer $igroup -dtype dedicated -z $Size
}

function Getigroup{
        $igroup = Read-Host "Please provide igroup"
        if ($AggregateName -eq $null) {
                Write-Output "igroup need to be provided!"
                Exit
        }
        return $AggregateName
}

#Get Information for 1st database

$DB1.DBName = GetDatabaseName 1 "Adventureworks"
$DB1.mdf = GetDatabaseFile "mdf" $DB1.DBName "Adventureworks.mdf"
$DB1.mdf_size = GetFileSize $DB1.mdf 5
$DB1.mdf_volname = GetVolumeName $DB1.mdf "Adventureworks_mdf"
$DB1.mdf_lunname = GetLunName $DB1.mdf "Adventureworks_mdf"

$DB1.ldf = GetDatabaseFile "ldf" $DB1.DBName "Adventureworks.ldf"
$DB1.ldf_size = GetFileSize $DB1.ldf 5
$DB1.ldf_volname = GetVolumeName $DB1.ldf "Adventureworks_ldf"
$DB1.ldf_lunname = GetLunName $DB1.ldf "Adventureworks_ldf"

#Get Information for 2nd database
$DB2.DBName = GetDatabaseName 1 "Pubs"
$DB2.mdf = GetDatabaseFile "mdf" $DB2.DBName "Pubs.mdf"
$DB2.mdf_size = GetFileSize $DB2.mdf 5
$DB2.mdf_volname = GetVolumeName $DB2.mdf "Pubs_mdf"
$DB2.mdf_lunname = GetLunName $DB2.mdf "Pubs_mdf"
```

Storage and Database Provisioning (ProvisioningSQL.ps1)

```
$DB2.ndf = GetDatabaseFile "ndf" $DB2.DBName "Pubs.ndf"
$DB2.ndf_size = GetFileSize $DB2.ndf 5
$DB2.ndf_volname = GetVolumeName $DB2.ndf "Pubs_ndf"
$DB2.ndf_lunname = GetLunName $DB2.ndf "Pubs_ndf"

$DB2.ldf = GetDatabaseFile "ldf" $DB2.DBName "Pubs.ldf"
$DB2.ldf_size = GetFileSize $DB2.ldf 5
$DB2.ldf_volname = GetVolumeName $DB2.ldf "Pubs_ldf"
$DB2.ldf_lunname = GetLunName $DB2.ldf "Pubs_ldf"



#Connect to Controller
$Controller = ConnectController

#Get AggregateName
$Aggregate = GetAggregateName

#Get igroup
$igroup = Getigroup

#Create Volumes
CreateVolume $DB1.mdf_volname $AggregateName $DB1.mdf_size
CreateVolume $DB1.ldf_volname $AggregateName $DB1.ldf_size

CreateVolume $DB2.mdf_volname $AggregateName $DB2.mdf_size
CreateVolume $DB2.ndf_volname $AggregateName $DB2.ndf_size
CreateVolume $DB2.ldf_volname $AggregateName $DB2.ldf_size

#Create LUNs
$Computer = Get-Content env:Computername
#Create LUNs for DB1
$MountPoint = "C:\MSSQL\" + $DB1.DBName + "\" + $DB1.mdf_lunname
$DB1.mdf_mountpoint = $MountPoint
```

Storage and Database Provisioning (ProvisioningSQL.ps1)

```
CreateLun $MountPoint, $Controller , $DB1.volname, $DB1.lunname, $Computer, $igroup, ($Size * 1.2)
$MountPoint = "C:\MSSQL\" + $DB1.DBName + "\" + $DB1.ldf_lunname
$DB1.ldf_mountpoint = $MountPoint
CreateLun $MountPoint, $Controller , $DB1.volname, $DB1.lunname, $Computer, $igroup, ($Size * 1.2)

#Create LUNs for DB2
$MountPoint = "C:\MSSQL\" + $DB2.DBName + "\" + $DB2.mdf_lunname
$DB2.mdf_mountpoint = $MountPoint
CreateLun $MountPoint, $Controller , $DB2.volname, $DB2.lunname, $Computer, $igroup, ($Size * 1.2)
$MountPoint = "C:\MSSQL\" + $DB2.DBName + "\" + $DB2.ndf_lunname
$DB2.ndf_mounpoint = $MountPoint
CreateLun $MountPoint, $Controller , $DB2.volname, $DB2.lunname, $Computer, $igroup, ($Size * 1.2)
$MountPoint = "C:\MSSQL\" + $DB2.DBName + "\" + $DB2.ldf_lunname
$DB2.ldf_mountpoint = $MountPoint
CreateLun $MountPoint, $Controller , $DB2.volname, $DB2.lunname, $Computer, $igroup, ($Size * 1.2)

#Create Database
#Create DB1
$cmd = "create database " + $DB1.DBName + " on (Name = " + $DB1.mdf_volname + ", filename = '" + $DB1.mdf_mountpoint
$cmd = $cmd + "\" + $DB1.mdf + "', SIZE = " + $DB1.mdf_size + ") "
$cmd = $cmd + "log on (Name = " + $DB1.ldf_volname + ", filename = '" + $DB1.ldf_mountpoint + "\" + $DB1.ldf + "', SIZE = " + $DB1.ldf_size + ")"
Invoke-sqlcmd -Query $cmd

#Create DB2
$cmd = "create database " + $DB2.DBName + " on (Name = " + $DB2.mdf_volname + ", filename = '" + $DB2.mdf_mountpoint
$cmd = $cmd + "\" + $DB2.mdf + "', SIZE = " + $DB2.mdf_size + "), "
$cmd = $cmd + "(Name = " + $DB2.ndf_volname + ", filename = '" + $DB2.ndf_mountpoint
$cmd = $cmd + "\" + $DB2.ndf + "', SIZE = " + $DB2.ndf_size + ") "
$cmd = $cmd + "log on (Name = " + $DB2.ldf_volname + ", filename = '" + $DB2.ldf_mountpoint + "\" + $DB2.ldf + "', SIZE = " + $DB2.ldf_size + ")"
Invoke-sqlcmd -Query $cmd
```

SMSQL Install (SMSQLInstall.ps1)

SMSQL Install (SMSQLInstall.ps1)

```
# +---------------------------------------------------------------------------
# | File : SMSQLInstall.ps1
# | Version : 1.00
# | Purpose : Installation of SnapManager for SQL
# | Synopsis:
# | Usage : .\SMSQLInstall.ps1
# +----------------------------------------------------------------------------
# | Maintenance History
# | -------------------
# | Name            Date   Version  C/R  Description
# | ----------------------------------------------------------------------------------
# | Mark Beaupre            2012-08-10   1.00    Borrowed heavily from the Pre-Provisioning.ps1 script
# +---------------------------------------------------------------------------------
# ******************************************************************************
$Account = Read-Host "Please provide SnapManager Service Account (Domain\Account)"
$Password = Read-Host "Please provide SnapManager Service Account Password"
$SMInstallDir = Read-Host "Please provide directory where SnapManager Installation Package is located"
#Install SnapManager
$cmd = $SMInstallDir + '\SMSQL5.2.exe /v /REBOOT=0 /L* ' + $SMInstallDir + '\SMInstall.log '
$cmd = $cmd + 'SVCUSERNAME=' + $Account + ' SVCUSERPASSWORD=' + $Password + '
SVCCONFIRMUSERPASSWORD=' + $Password
Invoke-Command $cmd
```

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

www.netapp.com

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexVol, RAID-DP, SnapDrive, SnapManager, SnapMirror, Snapshot, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Active Directory, Microsoft, SQL Server, Windows, and Windows Server are registered trademarks and Hyper-V and Windows PowerShell are trademarks of Microsoft Corporation. VMware is a registered trademark of VMware, Inc. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-4105-0912